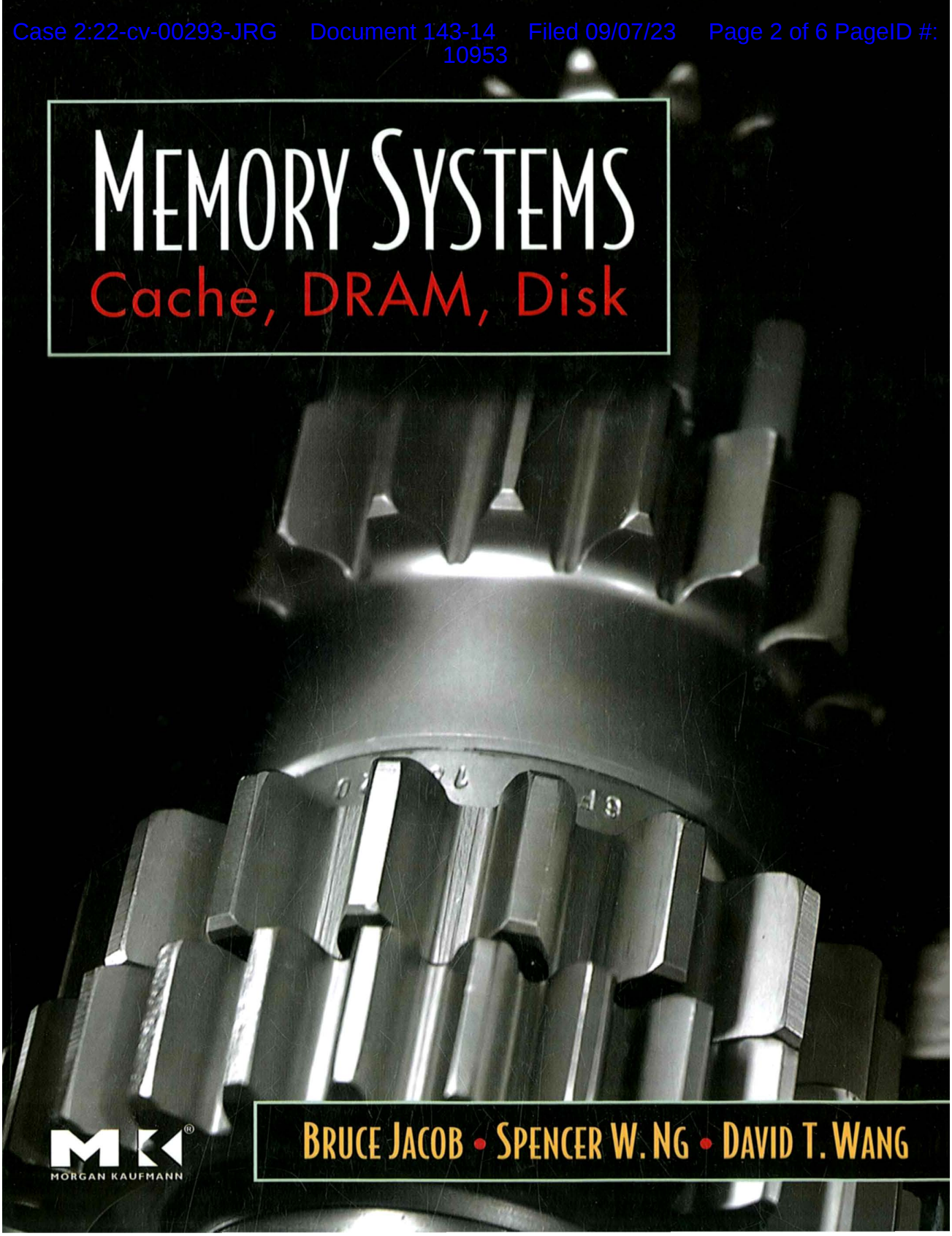
Exhibit N


MEMORY SYSTEMS
Cache, DRAM, Disk
BRUCE JACOB • SPENCER W. NG • DAVID T. WANG
MORGAN KAUFMANN

CHAPTER 7

Overview of DRAMs

DRAM is the "computer memory" that you order through the mail or purchase at the store. It is what you put more of into your computer as an upgrade to improve the computer's performance. It appears in most computers in the form shown in Figure 7.1—the ubiquitous *memory module*, a small computer board (a *printed circuit board*, or *PCB*) that has a handful of chips attached to it. The eight black rectangles on the pictured module are the DRAM chips: plastic packages, each of which encloses a *DRAM die* (a very thin, fragile piece of silicon).

Figure 7.2 illustrates DRAM's place in a typical PC. An individual DRAM device typically connects indirectly to a CPU (i.e., a microprocessor) through a memory controller. In PC systems, the memory controller is part of the *north-bridge* chipset that handles potentially multiple microprocessors, the graphics co-processor, communication to the *south-bridge* chipset (which, in turn, handles all of the system's I/O functions), as well as the interface to the DRAM system. Though still often referred to as "chipsets" these days, the north- and south-bridge chipsets are no longer sets of chips; they are usually implemented as single chips, and in some systems the functions of both are merged into a single die.

Because DRAM is usually an external device by definition, its use, design, and analysis must consider effects of implementation that are often ignored in the use, design, and analysis of on-chip memories such as SRAM caches and scratch-pads. Issues that a designer must consider include the following:

- Pins (e.g., their capacitance and inductance)
- Signaling
- Signal integrity
- Packaging
- Clocking and synchronization
- Timing conventions

Failure to consider these issues when designing a DRAM system is guaranteed to result in a sub-optimal, and quite probably non-functional, design.

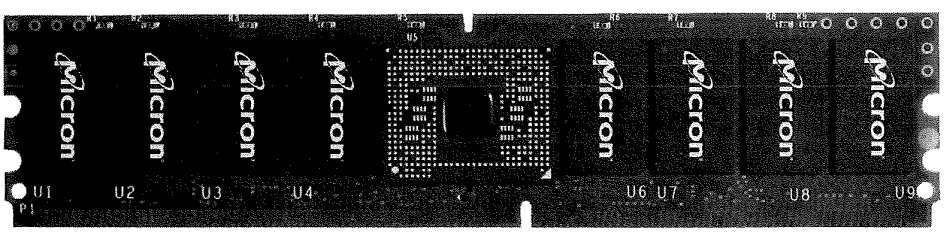


FIGURE 7.1: A memory module. A memory module, or DIMM (dual in-line memory module), is a circuit board with a handful of DRAM chips and associated circuitry attached to it.

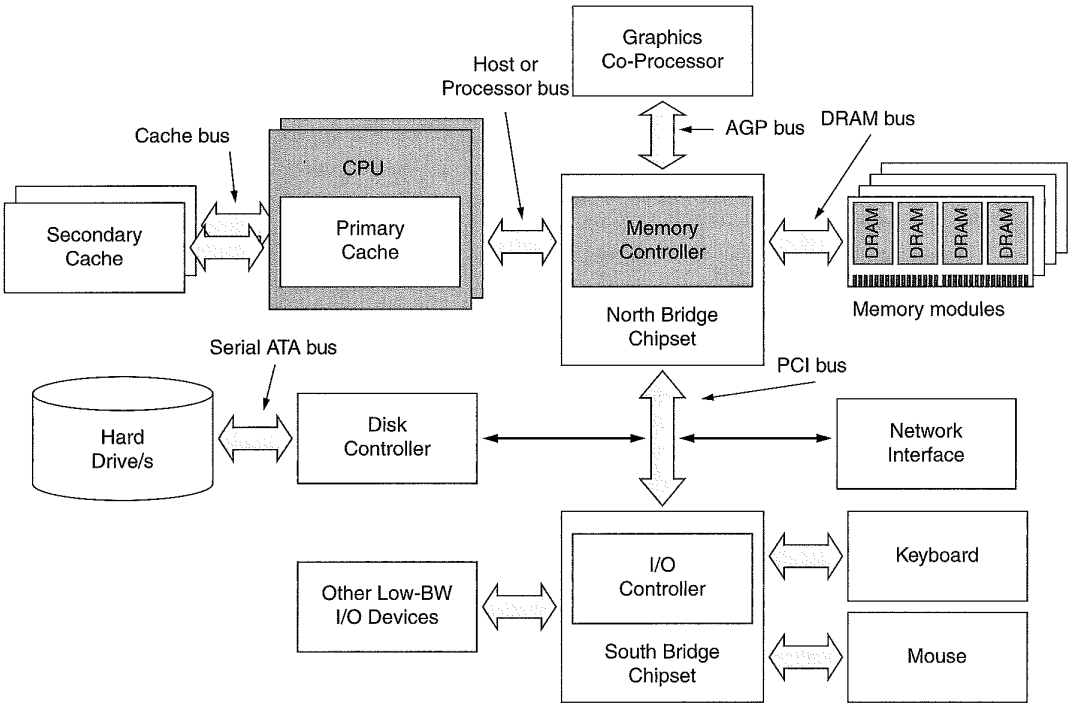


FIGURE 7.2: A typical PC organization. The DRAM subsystem is one part of a relatively complex whole. This figure illustrates a two-way multi-processor, with each processor having its own dedicated secondary cache. The parts most relevant to this report are shaded in darker grey: the CPU, the memory controller, and the individual DRAMs.

Thus, much of this section of the book deals with low-level implementation issues that were not covered in the previous section on caches.

7.1 DRAM Basics: Internals, Operation

A random-access memory (RAM) that uses a single transistor-capacitor pair for each bit is called a dynamic random-access memory or DRAM. Figure 7.3 shows, in the bottom right corner, the circuit for the storage cell in a DRAM. This circuit is *dynamic* because the capacitors storing electrons are not perfect devices, and their eventual leakage requires that, to retain information stored there, each capacitor in the DRAM must be periodically *refreshed* (i.e., read and rewritten).

Each DRAM die contains one or more *memory arrays*, rectangular grids of storage cells with each cell holding one bit of data. Because the arrays are rectangular grids, it is useful to think of them in terms associated with typical grid-like structures. A good example is a Manhattan-like street layout with avenues running north–south and streets running east–west. When one wants to specify a rendezvous location in such a city, one simply designates the intersection of a street and an avenue, and the location is specified without ambiguity. Memory arrays are organized just like this, except where Manhattan is organized into *streets* and *avenues*, memory arrays are organized

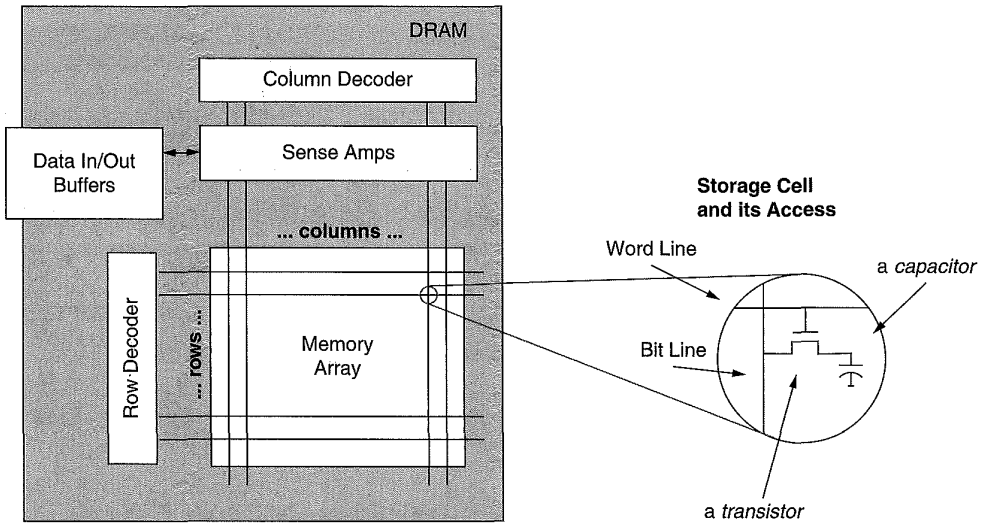


FIGURE 7.3: Basic organization of DRAM internals. The DRAM memory array is a grid of storage cells, where one bit of data is stored at each intersection of a *row* and a *column*.

into *rows* and *columns*. A DRAM chip's memory array with the rows and columns indicated is pictured in Figure 7.3. By identifying the intersection of a row and a column (by specifying a *row address* and a *column address* to the DRAM), a memory controller can access an individual storage cell inside a DRAM chip so as to read or write the data held there.

One way to characterize DRAMs is by the number of memory arrays inside them. Memory arrays within a memory chip can work in several different ways. They can act in unison, they can act completely independently, or they can act in a manner that is somewhere in between the other two. If the memory arrays are designed to act in unison, they operate as a unit, and the memory chip typically transmits or receives a number of bits equal to the number of arrays each time the memory controller accesses the DRAM. For example, in a simple organization, a x4 DRAM (pronounced "by four") indicates that the DRAM has at least four memory arrays and that a column width is 4 bits (each column read or write transmits 4 bits of data). In a x4 DRAM part, four arrays each read 1 data bit in unison, and the part sends out 4 bits of data each time the memory controller makes a column read request. Likewise, a x8 DRAM indicates that the DRAM has at least eight memory arrays and that a column width is 8 bits. Figure 7.4 illustrates the internal organization of x2, x4, and x8 DRAMs. In the past two decades, wider output DRAMs have appeared, and x16 and x32 parts are now common, used primarily in high-performance applications.

Note that each of the DRAM illustrations in Figure 7.4 represents multiple arrays but a single *bank*. Each set of memory arrays that operates independently of other sets is referred to as a bank, not an array. Each bank is independent in that, with only a few restrictions, it can be activated, precharged, read out, etc. at the same time that other banks (on the same DRAM device or on other DRAM devices) are being activated, precharged, etc. The use of multiple independent banks of memory has been a common practice in computer design since DRAMs were invented. In particular, *interleaving* multiple memory banks has been a popular method used to achieve high-bandwidth memory busses using

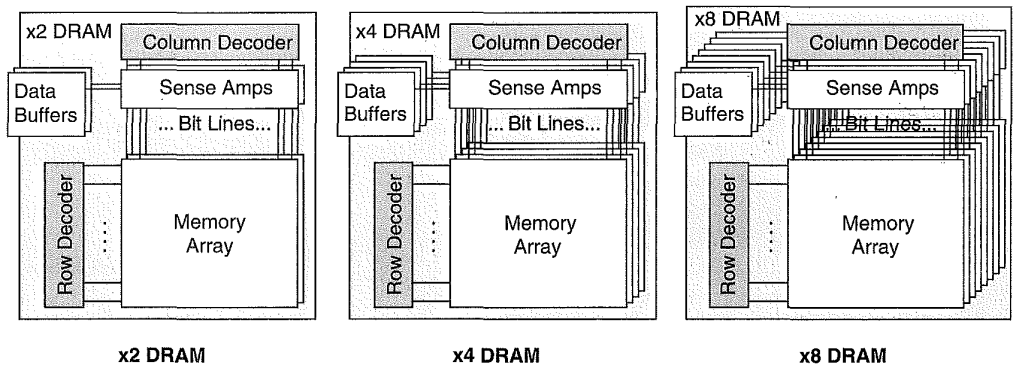


FIGURE 7.4: Logical organization of wide data-out DRAMs. If the DRAM outputs more than one bit at a time, the internal organization is that of multiple arrays, each of which provides one bit toward the aggregate data output.

low-bandwidth devices. In an interleaved memory system, the data bus uses a frequency that is faster than any one DRAM bank can support; the control circuitry toggles back and forth between multiple banks to achieve this data rate. For example, if a DRAM bank can produce a new chunk of data every 10 ns, one can toggle back and forth between two banks to produce a new chunk every 5 ns, or round-robin between four banks to produce a new chunk every 2.5 ns, thereby effectively doubling or quadrupling the data rate achievable by any one bank. This technique goes back at least to the mid-1960s, where it was used in two of the highest performance (and, as it turns out, best documented) computers of the day: the IBM System/360 Model 91 [Anderson et al. 1967] and Seymour Cray's Control Data 6600 [Thornton 1970].

Because a system can have multiple DIMMs, each of which can be thought of as an independent bank, and the DRAM devices on each DIMM can implement internally multiple independent banks, the word "rank" was introduced to distinguish DIMM-level independent operation versus internal-bank-level independent operation. Figure 7.5 illustrates the various levels of organization in a modern DRAM system. A system is composed of potentially many independent DIMMs. Each DIMM may contain one or more independent ranks. Each rank is a set of DRAM devices that operate in unison, and internally each of these DRAM devices implements one or more independent banks. Finally, each bank is composed of slaved memory arrays, where the number of arrays is equal to the data width of the DRAM part (i.e., a x4 part has four slaved arrays per bank). Having concurrency at the rank and bank levels provides bandwidth through the ability to pipeline requests. Having multiple DRAMs acting in unison at the rank level and multiple arrays acting in unison at the bank level provides bandwidth in the form of parallel access.

The busses in a JEDEC-style organization are classified by their function and organization into *data*, *address*, *control*, and *chip-select* busses. An example arrangement is shown in Figure 7.6, which depicts a memory controller connected to two memory modules. The data bus that transmits data to and from the DRAMs is relatively wide. It is often 64 bits wide, and it can be much wider in high-performance systems. A dedicated address bus carries row and column addresses to the DRAMs, and its width grows with the physical storage on a DRAM device (typical widths today are about 15 bits). A control bus is composed of the row and column strobes,[1] output enable, clock, clock enable, and other related signals. These

[1]A "strobe" is a signal that indicates to the recipient that another signal, e.g., data or command, is present and valid.